DOI: 10.22146/jnteti.v14i2.17117

Evaluasi Pengukuran Semantik Sinonim KBBI Menggunakan Pendekatan *Word Embedding*

Muhammad Rafli Aditya H.[1], Muhammad Ilham[1], Dewi Fatmarani Surianto[1*], Abdul Muis Mappalotteng[2]

[1] Departemen Teknik Informatika dan Komputer, Fakultas Teknik, Universitas Negeri Makassar, Makassar, Sulawesi Selatan 90224, Indonesia
[2] Departemen Pendidikan Teknik Elektro, Fakultas Teknik, Universitas Negeri Makassar, Makassar, Sulawesi Selatan 90224, Indonesia

[Diserahkan: 7 November 2024, Direvisi: 9 Februari 2025, Diterima: 16 April 2025]
Penulis Korespondensi: Dewi Fatmarani Surianto (email:dewifatmaranis@unm.ac.id)

INTISARI — Kamus Besar Bahasa Indonesia (KBBI) ialah salah satu sumber utama penyedia data dalam penelitian penentuan kemiripan makna kata dalam bahasa Indonesia. Penelitian ini membahas cara metode *word embedding* dan teknik pembobotan *term frequency-inverse document frequency* (TF-IDF) mengukur tingkat kemiripan pasangan makna kata sinonim untuk mengukur kemiripan pasangan makna kata sinonim dalam KBBI menggunakan *cosine similarity* dengan memanfaatkan teknik pembobotan TF-IDF dan beberapa model *word embedding* serta menerapkan *latent semantic analysis* (LSA). Metodologi penelitian ini dimulai dengan pengumpulan data, kemudian prapemrosesan teks yang terdiri atas *case folding*, *stopword removal*, *stemming*, dan *tokenization*. Selanjutnya, data yang telah diproses direpresentasikan ke dalam bentuk vektor menggunakan model *word embedding*, seperti Word2Vec, fastText, GloVe, *sentence - bidirectional encoder representations from transformers* (*Sentence-BERT*, S-BERT), dan teknik pembobotan TF-IDF. Lalu, LSA diterapkan untuk mereduksi dimensi vektor sebelum dilakukan uji kesamaan dengan *cosine similarity* dan diakhiri dengan evaluasi hasil. Hasil penelitian menunjukkan bahwa penggunaan fastText berhasil meningkatkan nilai kesamaan antara makna dua kata sinonim dengan nilai rata-rata yang diperoleh pada uji kesamaan dari 30 pasang makna kata sinonim adalah 0,901, dengan hasil evaluasi menunjukkan akurasi 0,88, *recall* 1,00, presisi 0,81, dan *F*1-*score* 0,90. Temuan ini menyimpulkan bahwa penggunaan fastText lebih efektif dalam meningkatkan akurasi pengukuran kemiripan makna kata sinonim. Rekomendasi untuk penelitian selanjutnya melibatkan perluasan korpus data dan eksplorasi lebih lanjut terhadap *word embedding* dalam uji kesamaan makna kata. Penelitian ini memberikan kontribusi pada pengembangan pemrosesan bahasa alami dan berpotensi menjadi dasar untuk aplikasi berbasis pemrosesan bahasa yang lebih akurat dalam mengukur kemiripan makna kata dalam KBBI.

KATA KUNCI — KBBI, Word2Vec, *Cosine Similarity*, fastText, GloVe, *Sentence-BERT*, Pengukuran Kesamaan Semantik.

I. PENDAHULUAN

Kamus Besar Bahasa Indonesia (KBBI) merupakan salah satu dari sumber utama dalam menentukan kemiripan makna kata untuk bahasa Indonesia. Kemiripan makna kata adalah kondisi dua kata atau lebih yang tidak sama, tetapi memiliki arti yang sama atau hampir serupa [1]. Kata yang dimaksud adalah sinonim karena menunjukkan kesamaan atau kemiripan makna dalam berbagai bentuk kata [2]. Mengenali kesamaan antara kata-kata merupakan langkah awal dalam menilai kesamaan antara kalimat, paragraf, dan dokumen. Kemiripan didefinisikan sebagai tingkat kesamaan antara dua potongan teks [3]. Untuk mengetahui kedekatan makna antarkalimat, dapat penghitungan kemiripan makna kalimat satu sama lain [4]. Kemiripan makna atau semantik merupakan konsep linguistik yang mengacu pada kondisi ketika dua kata berbeda dari sisi fonetik atau morfologi, tetapi memiliki arti yang sama atau mendekati kesamaan.

Pemrosesan bahasa alami *(natural language proccessing*, NLP) adalah cabang dari ilmu komputer dan kecerdasan buatan yang berfokus pada cara bahasa manusia alami dan komputer berinteraksi [5]. Dalam NLP, pengukuran kemiripan makna sangat penting untuk berbagai tugas. Misalnya, untuk mendapatkan terjemahan yang akurat, mesin penerjemah membutuhkan sistem untuk memahami frasa dan sinonim yang berbeda dari sisi morfologi dan fonetik, tetapi memiliki makna yang sama. Begitu pula dalam analisis sentimen, pengenalan kemiripan makna kata membantu sistem memahami perbedaan kata atau frasa yang digunakan untuk mengungkapkan emosi yang sama, sehingga ketika diterapkan pada aplikasi NLP dapat menghasilkan produk yang lebih akurat dan relevan [5].

Masalah yang diangkat pada penelitian ini adalah cara metode *word embedding* dan teknik pembobotan *term frequency-inverse document frequency* (TF-IDF) mengukur tingkat kemiripan pasangan makna kata sinonim untuk melihat metode yang paling efektif dalam uji kesamaan makna kata. Data yang digunakan pada penelitian ini adalah definisi setiap kata yang dipilih pada KBBI. Oleh karena itu, penelitian ini berfokus pada perhitungan kemiripan makna kata sinonim.

Terdapat berbagai metode komputasional yang telah dikembangkan dan dapat digunakan untuk mengukur kesamaan makna kata, mulai dari pendekatan berbasis aturan hingga metode pembelajaran mesin. Dalam penelitian ini, *cosine similarity* dipilih sebagai teknik utama untuk menghitung kemiripan dua *item*, atau pada penelitian ini disebut vektor yang mewakili kata-kata berdasarkan *embedding* kata-kata tersebut [6]. *Cosine similarity* dapat digunakan untuk menghitung kesamaan antarkalimat dan merupakan teknik populer dalam mengukur kesamaan teks [7]. Adapun beberapa model *word embedding* yang digunakan adalah Word2Vec, fastText, GloVe, dan *sentence - bidirectional encoder representations from transformers* (*Sentence-BERT*, S-BERT) yang dapat memproses data teks tidak terstruktur dengan mengambil kumpulan kata sebagai masukan dan menghasilkan vektor kata sebagai keluaran.

Word2Vec adalah metode atau algoritma dalam NLP yang digunakan untuk membuat representasi vektor dari kata-kata

(*word embeddings*) berdasarkan teks [8]. Dikembangkan oleh Mikolov dkk. pada 2013, model ini menggunakan jaringan saraf dengan lapisan tersembunyi untuk mengonversi kata ke dalam bentuk vektor padat. Keunggulannya terletak pada kemampuannya menangani sinonim dan homonim lebih baik dibandingkan representasi *sparse* tradisional [9], [10].

FastText adalah metode untuk mengekstraksi fitur dari kata dalam bentuk bilangan riil dengan menggunakan konsep *word embedding* berbasis prediksi. Metode ini merupakan pengembangan dari teknik *continuous bag of words* (CBOW) [11]. Keunggulan fastText terletak pada kemampuannya melatih model dengan cepat pada *dataset* besar dan menyediakan representasi untuk kata-kata yang tidak ada dalam data pelatihan. Jika sebuah kata tidak muncul selama pelatihan, kata tersebut dapat dipecah menjadi *n-gram* untuk menghasilkan *embedding vector*-nya [9].

Berbeda dengan Word2Vec dan fastText, GloVe adalah model berbasis hitungan yang mempelajari hubungan antarkata dengan menghitung frekuensi kemunculannya dalam korpus. Model ini menggunakan metode *global matrix factorization* untuk menangkap makna semantik secara lebih luas dibandingkan Word2Vec yang berbasis jaringan saraf [9]. Sementara itu, S-BERT merupakan modifikasi dari BERT yang menggunakan *siamese* dan *triplet networks* untuk membandingkan kesamaan semantik antarkalimat secara efisien. Dengan metode *pooling*, S-BERT mampu menghasilkan vektor tetap untuk kalimat, sehingga lebih cepat dibandingkan BERT dalam pencarian pasangan kalimat serupa [12]

Selain itu, penelitian ini juga menggunakan *latent semantic analysis* (LSA) untuk mereduksi dimensi vektor yang dihasilkan oleh *word embedding*. LSA adalah metode statistik yang digunakan untuk menentukan dan merepresentasikan kesamaan makna antara kata-kata dan teks melalui analisis terhadap teks dalam jumlah besar [13]. Diharapkan penelitian ini dapat meningkatkan akurasi pengukuran kemiripan makna kata dalam KBBI.

Beberapa penelitian sebelumnya telah mempelajari topik yang serupa, seperti membahas mengenai perhitungan kesamaan kalimat menggunakan *latent semantic indexing* (LSI) pada KBBI dengan nilai akurasi yang diperoleh 75,9% untuk kesamaan semantik menggunakan TF-IDF tradisional dan 80% untuk metode LSI dan TF-IDF [14]. Penelitian lain menggunakan fungsi terbilang dalam prapemrosesan dan *cosine similarity* dalam bahasa Indonesia untuk menguji kemiripan kalimat. Hasilnya menunjukkan bahwa 12 dari 13 pengujian (92,30%) mengalami peningkatan nilai kemiripan dibandingkan dengan prapemrosesan yang tidak menggunakan fungsi terbilang [15]. Selain itu, penelitian lain mengukur kemiripan makna menggunakan *cosine similarity* dan basis data sinonim kata, dengan hasil menunjukkan bahwa dari 25 pengujian, 24 nilai kemiripan mengalami peningkatan. Rata-rata nilai kemiripan pada penggunaan *ID* sebagai vektor hitung adalah 94,48%, sedangkan pada metode atau alur pembanding adalah 69,96% [16]. Beberapa penelitian tersebut menggunakan *cosine similarity* dan LSI, tetapi belum menggunakan Word2Vec, fastText, GloVe, S-BERT, dan LSA.

Penelitian sebelumnya telah menggunakan Word2Vec untuk melakukan klasifikasi topik Twitter menggunakan metode *random forest* dan fitur ekspansi Word2Vec. Penelitian tersebut menggunakan tiga korpus Word2Vec (*tweet*, berita, serta gabungan antara *tweet* dan berita) serta tiga variasi ekspansi fitur (*Top* 1, *Top* 5, dan *Top* 10) untuk mencari model terbaik. Model terbaik ditemukan dengan fitur *Top* 5, yang mencapai nilai akurasi 99,49% [17]. Selanjutnya, ada juga penelitian yang menganalisis sentimen tinjauan film menggunakan Word2Vec dan metode *long short term memory* (LSTM) *deep learning*. Hasilnya menunjukkan akurasi sebesar 88,17%, dengan ukuran *word vector* 100, dan akurasi terendah sebesar 85,86%, dengan ukuran *word vector* 500 [18]. Selain itu, terdapat penelitian mengenai analisis sentimen transportasi *online* dengan menggunakan ekstraksi fitur model Word2Vec dan algoritma *support vector machine* (SVM). Pengujian menunjukkan kinerja yang cukup baik dengan aplikasi Gojek memperoleh akurasi 87%, presisi 93%, dan *recall* 84%, sedangkan aplikasi Grab mendapatkan akurasi 82%, presisi 89%, dan *recall* 83% [19]. Berbeda dengan beberapa penelitian sebelumnya, sebuah penelitian menggunakan Word2Vec untuk memberikan rekomendasi lagu lintas bahasa berdasarkan lirik [20]. Hasilnya menemukan sepuluh judul lagu dengan kemiripan lirik terhadap lagu masukan, sedangkan pengujian parameter kata dengan TF-IDF serta kata dekat menghasilkan nilai rata-rata presisi sepuluh tertinggi untuk kata TF-IDF [20]. Beberapa penelitian tersebut menggunakan Word2Vec, tetapi memiliki metode dan objek yang berbeda dengan penelitian yang dilakukan.

Berbagai penelitian sebelumnya telah mengeksplorasi pengukuran kemiripan makna kata dalam bahasa Indonesia menggunakan berbagai metode. Beberapa di antaranya menggunakan pendekatan berbasis aturan, teknik *cosine similarity*, dan LSI dalam konteks mengukur kemiripan makna kata. Meskipun metode ini memiliki kelebihan masing-masing, penelitian terdahulu belum mengintegrasikan berbagai teknik *word embedding*, seperti Word2Vec, fastText, GloVe, dan S-BERT dalam satu kerangka kerja untuk melihat perbandingan kinerja yang lebih komprehensif.

Selanjutnya, meskipun penggunaan *word embedding* telah berkembang pesat dalam beberapa tahun terakhir, kebanyakan penelitian berfokus pada bahasa lain atau aplikasi berbeda, seperti analisis sentimen atau klasifikasi teks. Penggunaan metode *embedding* secara khusus untuk mengukur kesamaan makna kata dalam bahasa Indonesia masih jarang ditemukan. Hal ini menandakan bahwa ada potensi yang dapat dieksplorasi dalam penerapan teknik-teknik ini untuk bahasa Indonesia, terutama mengingat karakteristik linguistiknya yang unik [21].

Untuk mengatasi kesenjangan tersebut, penelitian ini berkontribusi dengan mengombinasikan berbagai teknik *word embedding* (Word2Vec, fastText, GloVe, dan S-BERT) dalam satu kerangka kerja guna mengevaluasi efektivitas relatifnya dalam mengukur kesamaan makna kata dalam bahasa Indonesia. Selain itu, penelitian ini juga menerapkan LSA untuk mereduksi dimensi vektor, sehingga meningkatkan kinerja dalam pengukuran kesamaan semantik. Lebih lanjut, penelitian ini menyediakan *dataset* berbasis KBBI yang berisi pasangan kata sinonim sebagai dasar untuk mengukur akurasi pendekatan yang diuji. Sebagai bentuk evaluasi, penelitian ini menggunakan metrik seperti akurasi, presisi, *recall*, dan *F*1-*score* guna memberikan analisis kuantitatif yang lebih jelas terhadap efektivitas metode yang digunakan. Dengan inovasi tersebut, penelitian ini diharapkan dapat memberikan kontribusi yang lebih baik terhadap pengembangan NLP dalam bahasa Indonesia, terutama dalam pengukuran kemiripan makna kata untuk berbagai aplikasi seperti analisis teks.

II. METODOLOGI

Penelitian ini terdiri atas tujuh tahap, yaitu pengumpulan data, prapemrosesan teks, pembobotan kata, penerapan *word*

embedding, reduksi dimensi vektor dengan LSA, pengujian kesamaan *cosine similarity,* hingga evaluasi terhadap hasil uji kesamaan.

A. PENGUMPULAN DATA

Dataset yang digunakan adalah hasil pengumpulan data yang dilakukan dengan mengumpulkan 300 makna kata sinonim dari *website* KBBI Kementerian Pendidikan dan Kebudayaan yang akan digunakan untuk uji kesamaan makna kata dengan dua pendekatan, yaitu *word embedding* dan teknik pembobotan TF-IDF yang ada pada penelitian ini. Pemilihan 300 data pada penelitian ini didasarkan pada pertimbangan efisiensi [22]. Jumlah tersebut diharapkan cukup untuk memberikan gambaran mengenai kinerja model Word2Vec dalam mengukur tingkat kesamaan makna kata. Penggunaan 300 data juga memungkinkan proses analisis dan evaluasi berjalan lebih optimal dalam hal waktu dan sumber daya komputasi untuk model yang dilatih. *Dataset* yang digunakan diperlihatkan pada Tabel I.

B. PRAPEMROSESAN TEKS

Prapemrosesan teks adalah serangkaian proses yang diterapkan pada dokumen teks sebelum penghitungan kemiripan. Proses ini bertujuan untuk membersihkan data dan mengonversi teks agar sesuai dengan standar yang dibutuhkan. Tahap ini memastikan data bersih dan siap digunakan secara optimal pada langkah-langkah selanjutnya. Secara umum, proses ini mencakup beberapa langkah [23] dan pada penelitian ini terdiri atas beberapa tahapan, yaitu *case folding* untuk mengatasi ketidaksesuaian seperti sensitivitas terhadap huruf besar akibat kapitalisasi yang tidak konsisten [24], *stopword removal* untuk menghapus kata-kata yang tidak memiliki makna dalam dokumen [25], *stemming* untuk mengembalikan kata-kata ke bentuk dasarnya [26], dan *tokenization* untuk memecah teks atau kalimat menjadi potongan kata yang disebut *token* [27].

C. PEMBOBOTAN KATA

TF-IDF digunakan untuk menetapkan bobot pada setiap kata dalam *dataset*. Proses ini melibatkan penerapan algoritma TF-IDF, yang memberikan skor berdasarkan frekuensi kata, dengan fokus pada kata-kata yang relevan, yaitu kata-kata yang sering muncul dalam satu dokumen, tetapi jarang muncul di dokumen lainnya [28]. Pada penelitian ini, pembobotan dilakukan menggunakan *library* TfidfVectorizer.

$$IDF = Log(\frac{n}{df_i}). \quad (1)$$

Persamaan (1) merupakan formula IDF, dengan n adalah jumlah kalimat dan df_i adalah jumlah kemunculan kata (*term*).

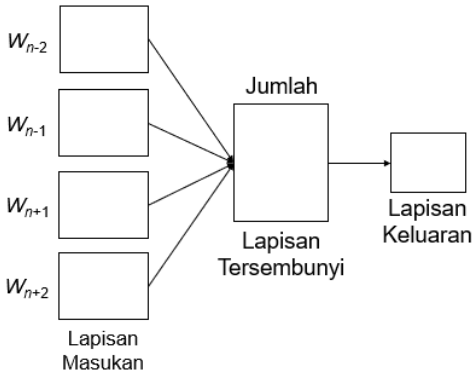
D. WORD EMBEDDING

Word embedding adalah istilah yang merujuk pada teknik pemodelan bahasa dan pembelajaran fitur dalam NLP. Setiap kata dalam kosakata direpresentasikan dengan vektor yang menggambarkan maknanya. Kata-kata tersebut dipetakan ke dalam bentuk vektor bilangan riil [29]. *Word embedding* yang digunakan dalam penelitian ini adalah Word2Vec, fastText, GloVe, dan S-BERT.

Word2Vec yang dikembangkan pada penelitian ini dibangun dengan menggunakan *dataset* yang berisi kumpulan definisi dari 300 kata di KBBI dan menggunakan arsitektur CBOW, seperti pada Gambar 1. CBOW merupakan model yang digunakan untuk menghasilkan representasi vektor kata. Dari arsitektur CBOW pada Gambar 1, tampak bahwa lapisan

TABEL I
DATASET MAKNA KATA SINONIM

Kata	Makna Kata
abrasi	pengikisan batuan oleh air; es; atau angin yg mengandung dan mengangkut hancuran bahan; luka lecet atau jejas karena pengikisan kulit oleh benda kasar; pengikisan selaput lendir (misalnya dalam membersihkan rahim)
pengikisan	Proses cara perbuatan mengikis; situasi saling mengikis; erosi
ayah	Orang tua kandung laki-laki; bapak; panggilan kepada orang tua kandung laki-laki.
...	...
...	...
pengabdian	Proses; cara; perbuatan mengabdi atau mengabdikan
dialog	Percakapan (dalam sandiwara; cerita; dan sebagainya): karya tulis yang disajikan dalam bentuk percakapan antara dua tokoh atau lebih
obrolan	Percakapan ringan dan santai; omong kosong



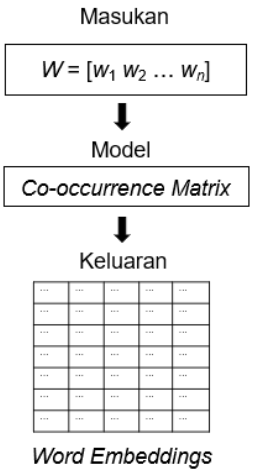
Gambar 1. Arsitektur *continuous bag of words.*

masukan berisi kata-kata konteks yang diambil dari sekitar kata target yang hendak diprediksi, lapisan tersembunyi berisi vektor dari kata-kata konteks yang dijumlahkan atau dirata-ratakan untuk membentuk suatu representasi vektor, sedangkan lapisan keluaran berisi representasi vektor yang dikalikan dengan bobot matriks untuk menghasilkan skor prediksi.

FastText pada penelitian ini menggunakan model pralatih untuk bahasa Indonesia bernama "cc.id.300.bin" yang dilatih pada Common Crawl dan Wikipedia. Model ini dilatih dengan menggunakan CBOW dalam dimensi sebesar 300, karakter *n-gram* dengan panjang 5, *window* berukuran 5, dan negatif 10. Pendekatan *n-gram* karakter berguna untuk menangkap informasi dari sisi morfologi, yang memungkinkan model untuk mempelajari representasi kata yang lebih baik, bahkan untuk kata-kata yang jarang atau tidak ada dalam korpus pelatihan.

GloVe pada penelitian ini menggunakan data pralatih bernama "glove_50dim_wiki.id.case.text" yang dilatih dengan data dari Wikipedia dan menghasilkan vektor berdimensi 50. Model ini menghasilkan *word embedding* yang menangkap hubungan semantik antara kata-kata berdasarkan statistik *co-occurrence* dalam korpus teks yang besar. *Co-occurrence* mengacu pada frekuensi kemunculan kata-kata secara bersama dalam suatu konteks tertentu. Arsitektur model GloVe ditunjukkan pada Gambar 2.

S-BERT pada penelitian ini menggunakan model pralatih bernama "distiluse base-multilingual-cased-v2" yang telah dilatih untuk berbagai bahasa, sehingga mendukung hingga 50 bahasa, seperti bahasa Korea, bahasa Arab, bahasa Jerman, dan



Gambar 2. Arsitektur model GloVe.

bahasa Russia, termasuk bahasa Indonesia. Model ini dirancang untuk menghasilkan representasi vektor yang dapat menangkap makna semantik dari kalimat dalam berbagai bahasa. Model ini menghasilkan vektor dengan panjang 512. Arsitektur S-BERT diperlihatkan pada Gambar 3.

Pada arsitektur model S-BERT dalam Gambar 3, Kalimat A dan Kalimat B adalah dua kalimat yang digunakan sebagai masukan, BERT merupakan tahap ketika kalimat-kalimat tersebut dimasukkan ke dalam dua model BERT yang identik dan menghasilkan vektor representasi untuk setiap *token*, *Mean Pooling* adalah tahap untuk mendapatkan satu vektor representasi kalimat (*u* untuk Kalimat A dan *v* untuk Kalimat B), dan yang terakhir kesamaan antara kedua kalimat diukur dengan *cosine similarity*.

Penerapan Word2Vec, fastText, GloVe, dan S-BERT pada penelitian ini digunakan untuk menghasilkan vektor dokumen dari dua dokumen spesifik ("*doc1*" dan "*doc2*") yang merupakan makna kata sinonim dan dilakukan pengukuran *cosine similarity* antara kedua vektor dokumen tersebut.

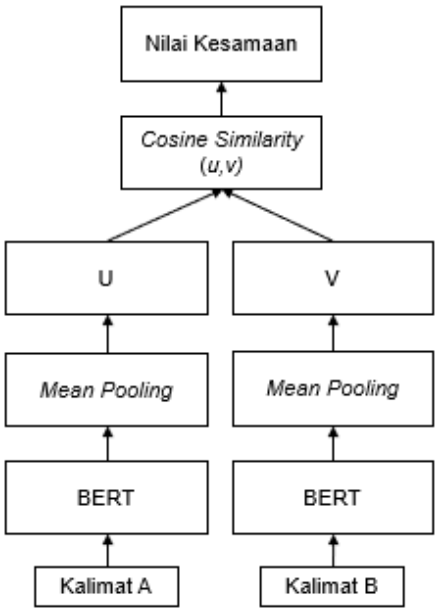
E. LATENT SEMANTIC ANALYSIS

LSA digunakan untuk mereduksi dimensi vektor hasil representasi dari *word embedding* seperti fastText, GloVe, dan S-BERT menggunakan TruncatedSVD dari sklearn. Teknik ini mendekomposisi matriks data besar menjadi berukuran lebih kecil dengan tetap mempertahankan informasi penting. Dalam penelitian ini, vektor dari fastText, GloVe, dan S-BERT direduksi menjadi 25 dimensi untuk meningkatkan efisiensi dalam pengukuran kesamaan makna kata. Word2Vec tidak mengalami reduksi karena modelnya telah dilatih dengan dimensi yang disesuaikan.

Reduksi dimensi dengan LSA membantu mengatasi *sparsity* dengan menyaring informasi yang lebih relevan, sehingga menghasilkan representasi kata yang lebih padat dan lebih terhubung secara semantik. Selain itu, LSA dapat mengidentifikasi hubungan laten antarkata, memungkinkan kata-kata yang jarang muncul tetap dihubungkan dengan kata-kata lain yang lebih umum dalam konteks semantik.

F. PENGUKURAN KESAMAAN

Cosine similarity adalah metode yang digunakan untuk mengukur kesamaan antara dua dokumen. Metode ini bekerja dengan mengukur kesamaan antara dua vektor dalam ruang dimensi, berdasarkan nilai kosinus dari sudut yang terbentuk oleh perkalian kedua vektor tersebut. Karena kosinus sudut 0 adalah 1 dan nilai kosinus untuk sudut lainnya kurang dari 1,



Gambar 3. Arsitektur model S-BERT.

nilai kesamaan antara dua vektor dianggap tinggi ketika nilai kosinusnya mendekati 1, yang menunjukkan bahwa kedua vektor sangat mirip [30]. Pada penelitian ini, *cosine similarity* berguna dalam menguji kemiripan makna kata sinonim berdasarkan hasil vektor dari *word embedding* dan TF-IDF.

$$\cos(\theta) = \frac{A.B}{\|A\|\,\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2 \times \sum_{i=1}^{n}(B_i)^2}}. \quad (2)$$

Persamaan (2) merupakan formula *cosine similarity*, dengan A dan B mengacu pada vektor yang akan digunakan dalam uji kemiripan, $A.B$ mengacu pada *dot product* antara vektor A dan B, $|A|$ dan $|B|$ mengacu pada panjang vektornya, dan $|A||B|$ mengacu pada *cross product* antara $|A|$ dan $|B|$.

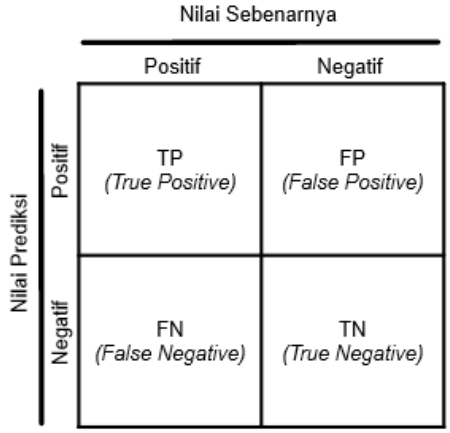
G. EVALUASI

Hasil pengukuran kesamaan atau *cosine similarity* diuji menggunakan empat metrik evaluasi, yaitu akurasi, *recall*, presisi, dan *F*1-*score*, terhadap 60 sampel. Sampel ini terdiri atas 30 hasil uji kesamaan pada makna kata bersinonim dan 30 hasil uji kesamaan pada makna kata yang tidak bersinonim.

Pada penelitian ini, evaluasi dilakukan dengan *threshold* 0,7 sebagai ambang batas untuk menentukan suatu pasangan kata dianggap bersinonim atau tidak. Nilai kesamaan yang diperoleh akan dikonversi menjadi label biner: jika nilai kesamaan >= 0,7, maka dikategorikan sebagai sinonim (label 1), sedangkan jika < 0,7, maka dikategorikan sebagai tidak sinonim (label 0).

Untuk membandingkan hasil prediksi dengan data referensi (*ground truth*), digunakan *confusion matrix*. *Ground truth* terdiri atas 30 angka 1 yang merepresentasikan pasangan kata bersinonim dan 30 angka 0 untuk pasangan kata yang tidak bersinonim. Kinerja model dalam mengklasifikasikan kata bersinonim kemudian dievaluasi dengan membandingkan hasil prediksi dengan *ground truth* menggunakan *confusion matrix*, yang memungkinkan penghitungan nilai akurasi, *recall*, presisi, dan *F*1-*score*.

Pada Gambar 4 terdapat beberapa istilah pada *confusion matrix*. *True negative* (TN) terjadi ketika model memprediksi data terdapat di kelas negatif dan sebenarnya data memang ada di kelas negatif, *true postive* (TP) terjadi ketika model



Gambar 4. *Confusion matrix.*

memprediksi data terdapat di kelas positif dan sebenarnya data memang ada di kelas positif, *false negative* (*FN*) terjadi ketika model memprediksi data terdapat di kelas negatif, tetapi sebenarnya data ada di kelas positif, sedangkan *false positive* (FP) terjadi ketika model memprediksi data terdapat di kelas positif , tetapi sebenarnya data ada di kelas negatif.

1) AKURASI

Akurasi merupakan rasio yang menunjukkan jumlah prediksi benar, baik positif maupun negatif, dibandingkan dengan keseluruhan data. Akurasi digunakan untuk mengukur ketepatan model dapat memprediksi. Akurasi dihitung menggunakan (3).

$$akurasi = \frac{TP+TN}{TP+TN+FP+FN}. \quad (3)$$

2) PRESISI

Presisi merupakan rasio yang menunjukkan perbandingan antara nilai TP dengan banyaknya data yang diprediksi positif. Presisi dihitung menggunakan (4).

$$presisi = \frac{TP}{TP+FP}. \quad (4)$$

3) *RECALL*

Recall merupakan rasio yang menunjukkan perbandingan antara nilai TP dengan banyaknya data yang sebenarnya positif. Persamaan (5) digunakan untuk menghitung *recall*.

$$recall = \frac{TP}{TP+FN}. \quad (5)$$

4) *F1-SCORE*

*F*1-*score* merupakan nilai yang menunjukkan kombinasi antara *recall* dan presisi. Rumus untuk memperoleh *F*1-*score* ditunjukkan pada (6).

$$F1-score = 2 \times \frac{presisi \times recall}{presisi+recall} \quad (6)$$

III. HASIL DAN PEMBAHASAN

Penelitian ini dimulai dengan melakukan pemrosesan teks pada dua makna sinonim kata di JupyterNotebook. Percobaan pertama menggunakan *library* TfidfVectorizer dan percobaan kedua menggunakan Word2Vec dengan *cosine similarity*. Hasil kedua percobaan tersebut lalu dibandingkan untuk mendapatkan metode yang menghasilkan nilai kesamaan tertinggi.

Sebelum tahap perhitungan TF-IDF dan *cosine similarity*, dilakukan prapemrosesan teks terlebih dahulu, dimulai dengan mengimpor dan menginisialisasi *library* dan data yang dibutuhkan. Data yang digunakan untuk uji kesamaan merupakan 30 pasang makna kata sinonim yang terdiri atas "ayah" dan "bapak", "ekonomis" dan "hemat", "datuk" dan "kakek", "kakak" dan "abang", "tabiat" dan "watak", "pelestarian" dan "konservasi", "giat" dan "rajin", "rumah" dan "hunian", "seniman" dan "artis", "guru" dan "pengajar", "pelajar" dan "siswa", "instruksi" dan "arahan", "bakat" dan "talenta", "penelitian" dan "riset", "ahli" dan "pakar", "berdikari" dan "mandiri", "berhasil" dan "sukses", "topan" dan "siklon", "swatantra" dan "otonomi", "flora" dan "tumbuhan", "kebisaan" dan "kepandaian", "vandalisme" dan "destruksi", "advokat" dan "pengacara", "wabah" dan "epidemi", "mampu" dan "bisa", "laris" dan "laku", "uzur" dan "halangan", "dedikasi" dan "pengabdian", "keahlian" dan "kepandaian", "dokter" dan "tabib". Beberapa kata sinonim ini dipilih karena sering digunakan dalam kehidupan sehari-hari, sehingga familier bagi masyarakat dan mudah dipahami dalam konteks pengujian kesamaan makna kata. Makna kata sinonim tersebut diambil dari KBBI seperti contoh sebagai berikut.

ayah = orang tua kandung laki-laki; bapak; panggilan kepada orang tua kandung laki-laki

bapak = orang tua laki-laki; ayah; orang laki-laki yang dalam pertalian kekeluargaan boleh dianggap sama dengan ayah (seperti saudara laki-laki ibu atau saudara laki-laki bapak); orang yang dipandang sebagai orang tua atau orang yang dihormati (seperti guru kepala kampung); panggilan kepada orang laki-laki yang lebih tua dari yang memanggil; orang yang menjadi pelindung (pemimpin, perintis jalan, dan sebagainya yang banyak penganutnya); pejabat

Adapun beberapa tahapan pemrosesan teks yang dilakukan adalah sebagai berikut. Pertama, *case folding* dilakukan untuk menyeragamkan teks menjadi huruf kecil tanpa kapital seperti berikut ini.

ayah = orang tua kandung laki-laki; bapak; panggilan kepada orang tua kandung laki-laki

bapak = orang tua laki-laki; ayah; orang laki-laki yang dalam pertalian kekeluargaan boleh dianggap sama dengan ayah (seperti saudara laki-laki ibu atau saudara laki-laki bapak); orang yang dipandang sebagai orang tua atau orang yang dihormati (seperti guru kepala kampung); panggilan kepada orang laki-laki yang lebih tua dari yang memanggil; orang yang menjadi pelindung (pemimpin perintis jalan dan sebagainya yang banyak penganutnya); pejabat

Kedua, dilakukan proses *stopword removal* untuk menghilangkan kata-kata yang tidak deskriptif atau tidak penting sehingga kalimat menjadi seperti berikut.

ayah = orang tua kandung laki-laki; bapak; panggilan orang tua kandung laki-laki

bapak = orang tua laki-laki; ayah; orang laki-laki pertalian kekeluargaan dianggap ayah (seperti saudara laki-laki saudara laki-laki bapak); orang dipandang orang tua orang dihormati (seperti guru kepala kampung); panggilan orang laki-laki tua memanggil; orang pelindung (pemimpin perintis jalan penganutnya); pejabat

Ketiga, proses *stemming* dilakukan untuk menghilangkan imbuhan atau menjadikan semua kata menjadi kata dasar, sehingga menjadi seperti berikut.

ayah = orang tua kandung laki bapak panggil orang tua kandung laki
bapak = orang tua laki ayah orang laki tali keluarga anggap ayah seperti saudara laki saudara laki bapak orang pandang orang tua orang hormat seperti guru kepala kampung panggil orang laki tua panggil orang lindung pimpin rintis jalan anut jabat

Setelah pemrosesan teks dilakukan, TF-IDF dan *cosine similarity* digunakan untuk menguji kesamaan antara dua makna kata sinonim sebagai percobaan pertama dengan hasil seperti disajikan pada Tabel II.

Dari contoh pasangan kata pertama, nilai *cosine similarity* dengan implementasi TF-IDF adalah 0,59. Selanjutnya, dilakukan percobaan kedua untuk menguji kesamaan antara dua makna kata sinonim menggunakan Word2Vec. Model Word2Vec dibangun dengan beberapa parameter, yaitu sebagai berikut. Parameter *preprocessed_data* adalah data yang berisi makna kata sinonim yang telah melalui prapemrosesan teks sebelumnya untuk digunakan dalam melatih model. Ukuran vektor merupakan jumlah dimensi vektor kata yang akan dihasilkan oleh model; pada penelitian ini, ukuran vektor yang digunakan bervariasi, seperti 10, 25, 50, 75, dan 100. *Window* atau jendela konteks merupakan jumlah kata sebelumnya dan berikutnya yang akan dipertimbangkan oleh model ketika melatih vektor kata; pada penelitian ini, ukuran *window* yang digunakan adalah 4, yang artinya empat kata dari sebelum dan sesudah kata target akan digunakan dalam memprediksi kata target atau melatih vektor kata. *min count* = 1 berarti jumlah minimum kemunculan suatu kata adalah satu kali yang akan dimasukkan dalam model. *workers* = 4 merupakan jumlah *thread* yang akan digunakan untuk pelatihan. *Skip-gram* (SG) merupakan parameter dalam menentukan model Word2Vec menggunakan arsitektur SG atau CBOW; penelitian ini menggunakan CBOW, sehingga parameter SG diberi nilai 0 karena jika diberi nilai 1, model akan menggunakan SG.

CBOW dipilih karena efisien secara komputasi, terutama saat digunakan untuk memproses *dataset* yang besar dan beragam. Dalam Word2Vec, CBOW digunakan untuk memprediksi kata target berdasarkan konteks sekitarnya, membuatnya lebih cepat untuk dilatih dibandingkan dengan menggunakan SG. Hal ini bermanfaat untuk uji kesamaan kata, terutama ketika *dataset* yang digunakan mencakup berbagai makna kata sinonim, seperti *dataset* dengan 300 makna kata yang digunakan pada penelitian ini. Pengujian ini dilakukan beberapa kali dengan nilai ukuran vektor yang berbeda-beda untuk mengetahui nilai kesamaan tertinggi pada setiap ukuran vektornya. Diperoleh beberapa nilai seperti ditunjukkan pada Tabel III.

Setelah itu, dilakukan pengujian berikutnya dengan beberapa *word embedding* lainnya, seperti fastText, GloVe, dan S-BERT, serta menggunakan LSA untuk mereduksi dimensi vektor menjadi 25, yang kemudian akan digunakan pada uji *cosine similarity*. Hasil uji kesamaan disajikan pada Tabel IV.

Skema tersebut selanjutnya diterapkan pada 29 pasang makna kata sinonim lainnya, sebagaimana definisi setiap kata yang diperoleh dari KBBI seperti pada Tabel V. Hasil rata-rata uji kesamaan dari Word2Vec dan *cosine similarity* pada 30 pasang makna kata sinonim dengan berbagai ukuran vektor yang berbeda-beda diperlihatkan pada Tabel VI.

Pengujian dilakukan pada 30 pasang makna kata sinonim dengan hasil rata-rata uji kesamaan yang berbeda untuk setiap ukuran vektornya. Ukuran vektor 10 menghasilkan nilai rata-rata kesamaan 0,484; ukuran vektor 25 menghasilkan nilai 0,504; ukuran vektor 50 menghasilkan nilai 0,505; ukuran vektor 75 menghasilkan nilai 0,478; dan ukuran vektor 100 menghasilkan nilai 0,495. Nilai rata-rata tertinggi dihasilkan oleh ukuran vektor 50, dengan nilai rata-rata 0,505. Sebagai perbandingan, Tabel VII menunjukkan rata-rata hasil uji kesamaan dari teknik pembobotan TF-IDF dan *word embedding* pada 30 pasang makna kata sinonim.

Berdasarkan hasil penelitian yang diperoleh, dapat dinyatakan bahwa uji kesamaan dengan TF-IDF dan beberapa *word embedding* menghasilkan nilai yang berbeda. Uji kesamaan dengan beberapa *word embedding* menghasilkan nilai kesamaan yang lebih tinggi dibandingkan dengan menggunakan TF-IDF. Hal ini terjadi karena adanya perbedaan

TABEL II
HASIL TF-IDF

Kata	Vektor TF-IDF	
	Ayah	***Bapak***
anggap	0,00	0,11
anut	0,00	0,11
ayah	0,00	0,23
bapak	0,21	0,08
guru	0,00	0,11
kandung	0,60	0,00
hormat	0,00	0,11
jabat	0,00	0,11
jalan	0,00	0,11
kampung	0,00	0,11
lindung	0,00	0,11
kepala	0,00	0,11
pimpin	0,00	0,11
rintis	0,00	0,11
pandang	0,00	0,11
keluarga	0,00	0,11
laki	0,42	0,42
orang	0,42	0,59
panggil	0,21	0,16
saudara	0,00	0,23
seperti	0,00	0,23
tali	0,00	0,11
tua	0,42	0,25

TABEL III
HASIL PENGUJIAN WORD2VEC DENGAN *COSINE SIMILARITY* DARI SINONIM AYAH/BAPAK

Ukuran Vektor	Hasil Uji Kesamaan
10	0,94
25	0,85
50	0,88
75	0,90
100	0,86

TABEL IV
HASIL PENGUJIAN *WORD EMBEDDING* DENGAN *LSA* DAN *COSINE SIMILARITY* PADA KATA AYAH DAN BAPAK

Kata	Hasil Uji Kesamaan		
	fastText	***GloVe***	***S-BERT***
ayah dan bapak	0,97	0,97	0,69

p-ISSN 2301–4156 | e-ISSN 2460–5719

TABEL V
MAKNA 30 PASANG KATA SINONIM

Kata	Makna Kata
ekonomis	bersifat hati-hati dalam pengeluaran uang; penggunaan barang; bahasa; waktu; tidak boros; hemat
hemat	berhati-hati dalam membelanjakan uang dan sebagainya; tidak boros; cermat
...	...
...	...
dokter	lulusan pendidikan kedokteran yang ahli dalam hal penyakit dan pengobatan
tabib	orang yang pekerjaannya mengobati orang sakit secara tradisional; seperti dukun; dokter

TABEL VI
HASIL RATA-RATA PENGUJIAN WORD2VEC DENGAN *COSINE SIMILARITY*

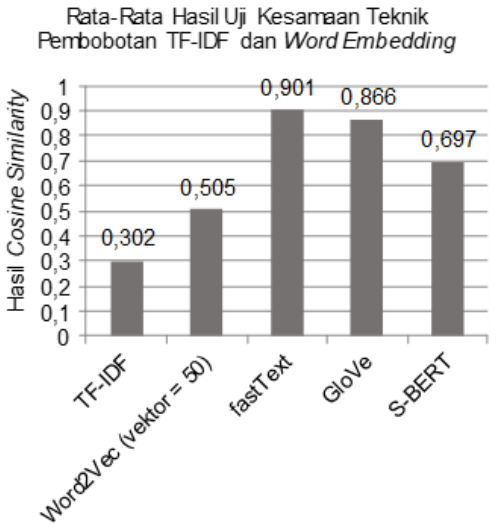
Ukuran Vektor	Hasil Rata-Rata Uji Kesamaan
10	0,484
25	0,504
50	0,505
75	0,478
100	0,495

TABEL VII
RATA-RATA HASIL PENGUJIAN TF-IDF DAN *WORD EMBEDDING* DENGAN *LSA* DAN *COSINE SIMILARITY* TERHADAP 30 PASANG MAKNA KATA

Kata	Rata-Rata Hasil Uji Kesamaan			
	fastText	***GloVe***	***S-BERT***	***TF-IDF***
30 pasang makna kata sinonim (sesuai dengan yang ada pada Tabel V)	0,901	0,866	0,697	0,302

dalam cara kedua metode tersebut merepresentasikan kata dan menangkap makna semantik kata atau kalimat. *Word embedding* seperti Word2Vec, GloVe, atau fastText mempelajari representasi kata berdasarkan konteks ketika kata tersebut digunakan atau muncul dalam kalimat. Model ini memahami hubungan antarkata dengan menangkap makna semantik, sehingga kata-kata yang memiliki makna serupa atau sinonim akan memiliki vektor yang dekat atau cukup mirip. Di sisi lain, S-BERT dirancang untuk memberikan representasi pada keseluruhan kalimat, sedangkan TF-IDF hanya melakukan perhitungan terhadap kemunculan atau frekuensi kata tanpa memperhitungkan konteks semantik, yang membuat metode tersebut menghasilkan nilai kesamaan yang cukup rendah jika dibandingkan dengan *word embedding* lainnya. Hasil rata-rata uji kesamaan 30 pasang makna kata sinonim dengan *word embedding* dan teknik pembobotan TF-IDF dapat dilihat pada Gambar 5.

Dari hasil uji kesamaan dengan *word embedding* dan teknik pembobotan TF-IDF pada Gambar 5, dapat disimpulkan bahwa fastText memberikan skor yang lebih tinggi dan lebih sesuai dengan makna sebenarnya serta menghasilkan nilai rata-rata tingkat kesamaan yang lebih tinggi dan stabil pada semua makna kata sinonim yang diuji coba jika dibandingkan dengan representasi vektor dari TF-IDF dan *word embedding* lainnya. Hasil rata-rata yang diperoleh fastText adalah 0,901, sedangkan nilai rata-rata yang diperoleh dengan pendekatan GloVe adalah 0,866, dengan selisih 0,035. Nilai rata-rata yang diperoleh dengan pendekatan S-BERT adalah 0,697, dengan selisih 0,204; nilai rata-rata tertinggi yang diperoleh dengan pendekatan Word2Vec adalah 0,505, dengan selisih 0,396; dan



Gambar 5. Grafik hasil *cosine similarity* dengan TF-IDF dan *word embedding* pada uji kesamaan makna kata sinonim.

nilai rata-rata yang diperoleh dengan pendekatan TF-IDF adalah 0,302, dengan selisih 0,599.

Namun, hasil pengukuran kesamaan ini belum cukup untuk memvalidasi efektivitas metode *word embedding* yang digunakan. Oleh karena itu, dilakukan evaluasi lebih lanjut menggunakan empat metrik pengujian, yaitu akurasi, *recall*, presisi, dan *F1-score*. Pengujian ini dilakukan dengan membandingkan hasil pengukuran pada 60 sampel, yang terdiri atas 30 pasangan kata bersinonim dan 30 pasangan kata tidak bersinonim. Daftar pasangan kata sinonim disajikan dalam Tabel V, sedangkan pasangan kata tidak bersinonim dibentuk secara acak, seperti "ayah" dan "abang", "kakak" dan "bapak", "datuk" dan "hemat", "ekonomis" dan "kakek", "tabiat" dan "konservasi", "pelestarian" dan "watak", "giat" dan "hunian", "rumah" dan "rajin", "guru" dan "artis", "seniman" dan "pengajar", "pelajar" dan "arahan", "instruksi" dan "siswa", "bakat" dan "riset", "penelitian" dan "talenta", "ahli" dan "mandiri", "berdikari" dan "pakar", "berhasil" dan "siklon", "topan" dan "sukses", "swatantra" dan "tumbuhan", "flora" dan "otonomi", "kebisaan" dan "destruksi", "vandalisme" dan "kepandaian", "advokat" dan "epidemi", "wabah" dan "pengacara", "mampu" dan "laku", "laris" dan "bisa", "uzur" dan "pengabdian", "dedikasi" dan "halangan", "keahlian" dan "tabib", serta "dokter" dan "kepandaian".

Keakuratan metode *cosine similarity* diukur dengan melihat kemampuan model dalam membedakan kata-kata yang benar-benar sinonim dari yang tidak. Jika metode ini memberikan nilai kesamaan tinggi pada pasangan kata yang tidak mirip, hasil dianggap kurang baik. Sebaliknya, jika *cosine similarity* mampu secara konsisten memberikan skor kesamaan tinggi pada kata-kata yang memang sinonim, metode yang digunakan dapat dianggap efektif.

Hasil yang didapatkan adalah *cosine similarity* dengan TF-IDF memperoleh nilai akurasi 0,50, presisi 0,00, *recall* 0,00, dan *F1-score* 0,00. Di pendekatan lainnya, Word2Vec memperoleh nilai akurasi 0,60, presisi 0,88, *recall* 0,23, dan *F1-score* 0,37; fastText memperoleh nilai akurasi 0,88, presisi 0,81, *recall* 1,00, dan *F1-score* 0,90; GloVe memperoleh nilai akurasi 0,80, presisi 0,74, *recall* 0,93, dan *F1-score* 0,82; dan S-BERT memperoleh nilai akurasi 0,77, presisi 1,00, *recall* 0,53, dan *F1-score* 0,70. Hasil lengkapnya disajikan pada Tabel VIII.

TABEL VIII
HASIL EVALUASI TERHADAP TF-IDF DAN *WORD EMBEDDING*

Metode	Hasil Evaluasi			
	Akurasi	*Recall*	*Presisi*	*F1-Score*
TF-IDF	0,50	0,00	0,00	0,00
Word2Vec	0,60	0,23	0,88	0,37
FastText	0,88	1,00	0,81	0,90
GloVe	0,80	0,93	0,74	0,82
S-BERT	0,77	0,53	1,00	0,70

Hasil yang berbeda-beda tersebut terjadi karena terdapat perbedaan pada cara merepresentasikan kata dan *dataset* yang digunakan dari setiap metode pada penelitian ini. Misalnya, Word2Vec yang digunakan merepresentasikan kata ke dalam bentuk vektor berdasarkan konteks lokal yang ada di sekitar kata dengan menggunakan CBOW untuk menghasilkan representasi vektor kata. Namun, metode ini tidak memperhitungkan konteks global yang lebih luas karena hanya berfokus pada konteks lokal di sekitar kata target dan *dataset* yang digunakan hanya berisi 300 makna kata sinonim. Hal ini membuat Word2Vec memiliki keterbatasan dalam menangkap hubungan antarkata dalam skala yang lebih besar. Kemudian, GloVe dirancang untuk menangkap informasi *co-occurrence* (kemunculan) dari kata-kata di seluruh korpus dengan membuat matriks berisi data *co-occurrence* dengan menghitung kemunculan kata beberapa kali pada konteks tertentu. GloVe memanfaatkan statistik *co-occurrence* global, sehingga mampu menangkap hubungan semantik secara global dan lebih baik dalam menangkap konteks kata di seluruh dokumen. Namun, GloVe tidak dapat menangani kata-kata yang jarang atau tidak dikenal karena, seperti Word2Vec, GloVe tidak mempertimbangkan struktur subkata. Selanjutnya, TF-IDF hanya memberikan representasi vektor dari perhitungan sederhana dengan menghitung kemunculan atau frekuensi setiap kata yang ada pada kalimat atau dokumen, sehingga memiliki kekurangan dalam menangkap hubungan antarkata, tidak mempertimbangkan makna kata dalam konteks yang lebih luas, dan hasilnya sering kali tidak akurat.

Berbeda dengan *word embedding* lainnya, S-BERT dirancang untuk menghasilkan representasi vektor dari kalimat atau teks secara keseluruhan menggunakan *Siamese networks*, yang menghasilkan vektor berukuran tetap. Kekurangannya adalah metode ini memerlukan sumber daya komputasi yang lebih besar dibandingkan dengan teknik *word embedding* lainnya, seperti Word2Vec atau TF-IDF, dan juga tidak dirancang untuk menangkap makna semantik kata-kata individual karena fokus utama S-BERT adalah menghasilkan representasi keseluruhan kalimat atau teks yang lebih panjang, bukan pada kata-kata secara terpisah dan juga tidak mempertimbangkan struktur subkata.

Di sisi lain, pendekatan fastText merupakan hasil pengembangan dari Word2Vec, dengan keunggulan yaitu memanfaatkan subkata (*n-grams*) untuk menghasilkan representasi kata, yang membuat fastText dapat menangkap informasi morfologis dan mengatasi kata-kata yang jarang atau kata-kata baru, sehingga meningkatkan pemahaman model tentang hubungan semantik antarkata. Keunggulan ini memungkinkan fastText untuk memberikan representasi kata yang lebih kaya atau memperoleh lebih banyak informasi semantik dan morfologis, sehingga membuat fastText dapat memberikan nilai rata-rata tertinggi dan lebih stabil jika dibandingkan dengan *word embedding* lainnya.

Penelitian ini memberikan kontribusi dalam meningkatkan pemahaman penerapan berbagai metode NLP dalam konteks bahasa Indonesia, khususnya pada teknik TF-IDF, Word2Vec, fastText, GloVe, dan S-BERT. Hasil penelitian ini dapat menjadi dasar bagi penelitian lebih lanjut dalam meningkatkan teknik NLP yang lebih sesuai dengan karakteristik bahasa Indonesia. Selain itu, penelitian ini dapat meningkatkan kinerja berbagai aplikasi NLP, seperti analisis sentimen, klasifikasi teks, dan pengenalan suara (*speech recognition*).

Penelitian ini juga menekankan pentingnya mempertimbangkan konteks budaya dan bahasa lokal dalam pengembangan teknologi NLP. Dengan memahami cara model berfungsi dalam bahasa Indonesia, hasil penelitian ini dapat membantu praktisi dan pengembang NLP memilih metode *word embedding* yang paling sesuai untuk proyek tertentu. Misalnya, pemilihan metode yang optimal untuk aplikasi dengan keterbatasan sumber daya atau aplikasi yang memerlukan representasi semantik yang lebih mendalam.

Kontribusi lain dari penelitian ini adalah potensi perbaikannya terhadap berbagai aplikasi NLP berbasis bahasa Indonesia, seperti meningkatkan akurasi sistem penerjemahan mesin dan analisis teks berbasis konteks. Selain itu, penelitian ini juga berperan dalam pengembangan teknik NLP yang lebih efisien dan relevan, khususnya dalam uji kesamaan makna kata.

IV. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa pendekatan *word embedding* dengan fastText dan penerapan LSA berhasil meningkatkan kinerja *cosine similarity* dalam mengukur kesamaan makna kata berdasarkan KBBI. Metode ini terbukti lebih efektif dibandingkan dengan TF-IDF serta beberapa *word embedding* lainnya, seperti Word2Vec, GloVe, dan S-BERT. FastText menunjukkan hasil terbaik dalam menangkap kesamaan makna kata dengan rata-rata skor *cosine similarity* sebesar 0,901. Selain itu, hasil evaluasi metrik menunjukkan bahwa fastText memiliki akurasi sebesar 0,88, *recall* 1,00, presisi 0,81, dan *F*1-*score* 0,90. Keunggulan ini disebabkan oleh kemampuannya dalam memahami representasi kata yang lebih kontekstual dibandingkan metode lainnya. Untuk penelitian selanjutnya, disarankan agar cakupan korpus data diperluas dan dilakukan eksplorasi terhadap metode *word embedding* lainnya guna meningkatkan akurasi dan efektivitas pada pengukuran kesamaan makna kata dalam bahasa Indonesia.

KONFLIK KEPENTINGAN

Para penulis menyatakan bahwa tidak terdapat konflik kepentingan.

KONTRIBUSI PENULIS

Konseptualisasi, Muhammad Rafli Aditya H., Muhammad Ilham, dan Dewi Fatmarani Surianto; metodologi, Muhammad Rafli Aditya H. dan Muhammad Ilham; perangkat lunak, Muhammad Rafli Aditya H.; validasi data, Muhammad Rafli Aditya H., Muhammad Ilham, dan Dewi Fatmarani Surianto; analisis formal, Muhammad Rafli Aditya H. dan Dewi Fatmarani Surianto; investigasi, Muhammad Rafli Aditya H. dan Dewi Fatmarani Surianto; sumber daya, Muhammad Rafli Aditya H., Muhammad Ilham, dan Dewi Fatmarani Surianto; kurasi data, Muhammad Rafli Aditya H.; penulisan—penyusunan draf asli, Muhammad Rafli Aditya H., Muhammad Ilham, dan Dewi Fatmarani Surianto; penulisan—peninjauan dan penyuntingan, Muhammad Rafli Aditya H., Dewi Fatmarani Surianto, dan Abdul Muis Mappalotteng; visualisasi,

Muhammad Rafli Aditya H.; pengawasan, Dewi Fatmarani Surianto dan Abdul Muis Mappalotteng; administrasi proyek, Muhammad Rafli Aditya H. dan Dewi Fatmarani Surianto; akuisisi pendanaan, Muhammad Rafli Aditya H., Muhammad Ilham, Dewi Fatmarani Surianto, dan Abdul Muis Mappalotteng.

REFERENSI

[1] Y. Caterina, M.A. Yaqin, dan S. Zaman, "Pengukuran kemiripan makna kalimat dalam bahasa Indonesia menggunakan metode path," *Fountain Inform. J.*, vol. 6, no. 2, hal. 45–50, Nov. 2021, doi: 10.21111/fij.v6i2.4844.

[2] N.P. Paino, D.D.S. Hutagaol, dan A.U. Sagala, "Analisis penanda hubungan sinonim dan hiponimi pada puisi 'Membaca Tanda-Tanda' karya Taufiq Ismail," *Pena Literasi, J. Pendidik. Bhs. Sastra Indones.*, vol. 4, no. 1, hal. 37–44, Apr. 2021, doi: 10.24853/pl.4.1.37-44.

[3] J. Wang dan Y. Dong, "Measurement of text similarity: A survey," *Information*, vol. 11, no. 9, hal. 1–17, Sep. 2020, doi: 10.3390/info11090421.

[4] G.U. Abriani dan M.A. Yaqin, "Implementasi metode semantic similarity untuk pengukuran kemiripan makna antar kalimat," *ILKOMNIKA, J. Comput. Sci. Appl. Inform.*, vol. 1, no. 2, hal. 47–57, Des. 2019, doi: 10.28926/ilkomnika.v1i2.15.

[5] R.M. Arrasyid, D.E. Putera, dan A.Y.P. Yusuf, "Analisis sentimen review pembelian produk di marketplace Shopee menggunakan pendekatan natural language processing," *J. Tekno Kompak*, vol. 18, no. 2, hal. 319–330, Agu. 2024, doi: 10.33365/jtk.v18i2.3813.

[6] S.A. Zulvian, K. Prihandani, dan A.A. Ridha, "Perbandingan metode MSD dan cosine similarity pada sistem rekomendasi dengan pendekatan item-based collaborative filtering," *Intecoms, J. Inf. Technol. Comput. Sci.*, vol. 4, no. 2, hal. 340–347, Agu. 2021, doi: 10.31539/intecoms.v4i2.2781.

[7] Rismayani dkk., "Implementasi algoritma text mining dan cosine similarity untuk desain sistem aspirasi publik berbasis mobile," *Komputika, J. Sist. Komput.*, vol. 11, no. 2, hal. 169–176, Okt. 2022, doi: 10.34010/komputika.v11i2.6501.

[8] Y.A. Pradana, I. Cholissodin, dan D. Kurnianingtyas, "Analisis sentimen pemindahan Ibu Kota Indonesia pada media sosial Twitter menggunakan metode LSTM dan Word2Vec," *JPTIIK (J. Pengemb. Teknol. Inf. Ilmu Komput.)*, vol. 7, no. 5, hal. 2389–2397, Mei 2023.

[9] A. Nurdin, B.A.S. Aji, A. Bustamin, dan Z. Abidin, "Perbandingan kinerja word embedding Word2Vec, GloVe, dan fastText pada klasifikasi teks," *J. Tekno Kompak*, vol. 14, no. 2, hal. 74–79, Agu. 2020, doi: 10.33365/jtk.v14i2.796.

[10] R.P. Nawangsari, R. Kusumaningrum, dan A. Wibowo, "Word2Vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study," *Procedia Comput. Sci.*, vol. 157, hal. 360–366, Sep 2019, doi: 10.1016/j.procs.2019.08.178.

[11] R.P. Hastuti, V. Riona, dan M. Hardiyanti, "Content retrieval dengan fastText word embedding pada learning management system olimpiade," *J. Internet Softw. Eng.*, vol. 4, no. 1, hal. 18–22, Mei 2023, doi: 10.22146/jise.v4i1.6766.

[12] B. Juarto dan A.S. Girsang, "Neural collaborative with sentence BERT for news recommender system," *JOIV, Int. J. Inform. Vis.*, vol. 5, no. 4, hal. 448–455, Des. 2021, doi: 10.30630/joiv.5.4.678.

[13] L. Cagliero, P. Garza, dan E. Baralis, "ELSA: A multilingual document summarization algorithm based on frequent itemsets and latent semantic analysis," *ACM Trans. Inf. Syst. (TOIS)*, vol. 37, no. 2, hal. 1–33, Apr. 2019, doi: 10.1145/3298987.

[14] M. Panji M dan A.F. Huda, "Calculating the similarity of Indonesian sentences using latent semantic indexing based on KBBI," dalam *2022 Int. Conf. Inform. Multimed. Cyber Inf. Syst. (ICIMCIS)*, 2022, hal. 148–153, doi: 10.1109/ICIMCIS56303.2022.10017797.

[15] A. Sanjaya dan S.D. Sasongko, "Uji kemiripan kalimat menggunakan fungsi terbilang pada pre-processing dan cosine similarity dalam bahasa Indonesia," *NERO (Netw. Eng. Res. Oper.)*, vol. 7, no. 2, hal. 95–104, Nov. 2022.

[16] A. Sanjaya dkk., "Pengukuran kemiripan makna menggunakan cosine similarity dan basis data sinonim kata," *J. Teknol. Inf. Ilmu Komput.*, vol. 10, no. 4, hal. 747–752, Agu. 2023, doi: 10.25126/jtiik.2023106864.

[17] R.G. Ramli dan Y. Sibaroni, "Klasifikasi topik Twitter menggunakan metode random forest dan fitur ekspansi Word2Vec," *e-Proc, Eng.*, vol. 9, no. 1, hal. 79–92, Feb. 2022.

[18] W. Widayat, "Analisis sentimen movie review menggunakan Word2Vec dan metode LSTM deep learning," *J. Media Inform. Budidarma*, vol. 5, no. 3, hal. 1018–1026, Jul. 2021, doi: 10.30865/mib.v5i3.3111.

[19] E. Suryati, Styawati, dan A.A. Aldino, "Analisis sentimen transportasi online menggunakan ekstraksi fitur model Word2Vec text embedding dan algoritma support vector machine (SVM)," *J. Teknol. Sist. Inf.*, vol. 4, no. 1, hal. 96–106, Mar. 2023, doi: 10.33365/jtsi.v4i1.2445.

[20] G.W. Aldiansyah, P.P. Adikara, dan R.C. Wihandika, "Rekomendasi lagu cross language berdasarkan lirik menggunakan Word2Vec," *JPTIIK (J. Pengemb. Teknol. Inf. Ilmu Komput.),* vol. 3, no. 8, hal. 8036–8041, Agu. 2019.

[21] R. Julistiana, "Kosa kata bahasa Indonesia yang unik dan menarik," *Abdima Dejurnal*, vol. 1, no. 1, hal. 106–112, Apr. 2024.

[22] X. Rong, "Word2Vec parameter learning explained," 2014, *arXiv: 1411.2738*.

[23] H. Arfandy dan I.A. Musdar, "Rancang bangun sistem cerdas pemberian nilai otomatis untuk ujian esai menggunakan algoritma cosine similarity," *Inspir., J. Teknol. Inf. Komun.*, vol. 10, no. 2, hal. 123–136, Des. 2020.

[24] A.E. Sari, S. Widowati, dan K.M. Lhaksmana, "Klasifikasi ulasan pengguna aplikasi mandiri online di Google Play Store dengan menggunakan metode information gain dan naive Bayes classifier," *e-Proc. Eng.*, vol. 6, no. 2, hal. 9143–9157, Agu. 2019.

[25] R.S. Amardita, Adiwijaya, dan M.D. Purbolaksono, "Analisis sentimen terhadap ulasan Paris Van Java Resort Lifestyle Place di Kota Bandung menggunakan algoritma KNN," *JURIKOM (J. Ris. Komput.)*, vol. 9, no. 1, hal. 62–68, Feb. 2022, doi: 10.30865/jurikom.v9i1.3793.

[26] S. Lumbansiantar, S. Dwiasnati, dan N.S. Fatonah, "Penerapan metode cosine similarity dalam mendeteksi plagiarisme pada jurnal," *Format, J. Ilm. Tek. Inform.*, vol. 12, no. 2, hal. 142–150, Jul. 2023, doi: 10.22441/format.2023.v12.i2.007.

[27] Apriani, H. Zakiyudin, dan K. Marzuki, "Penerapan algoritma cosine similarity dan pembobotan TF-IDF system penerimaan mahasiswa baru pada kampus swasta," *J. Bumigora Inf. Technol. (BITe)*, vol. 3, no. 1, hal. 19–27, Jun. 2021, doi: 10.30812/bite.v3i1.1110.

[28] A.B.P. Negara, H. Muhardi, dan I.M. Putri, "Analisis sentimen maskapai penerbangan menggunakan metode naive Bayes dan seleksi fitur information gain," *J. Teknol. Inf. Ilmu Komput.*, vol. 7, no. 3, hal. 599–606, Jun. 2020, doi: 10.25126/jtiik.202071947.

[29] I.K.B.A.W. Kencana dan W. Maharani, "Klasifikasi opini pada fitur produk berbasis graph," *e-Proc. Eng.*, vol. 4, no. 2, hal. 3148–3155, Agu. 2017.

[30] M.D.R. Wahyudi, "Penerapan algoritma cosine similarity pada text mining terjemah Al-Qur'an berdasarkan keterkaitan topik," *Semesta Tek.*, vol. 22, no. 1, hal. 41–50, Mei 2019, doi: 10.18196/st.221235.